Towards Explainable NLP: A Generative Explanation Framework for Text Classification

Hui Liu[1], **Qingyu Yin**[2], **William Yang Wang**[3]
[1] Peking University, China
[2] Harbin Institute of Technology, China
[3] University of California, Santa Barbara, USA
layneliuhui@gmail.com
qyyin@ir.hit.edu.cn
william@cs.ucsb.edu

Abstract

Building explainable systems is a critical problem in the field of Natural Language Processing (NLP), since most machine learning models provide no explanations for the predictions. Existing approaches for explainable machine learning systems tend to focus on interpreting the outputs or the connections between inputs and outputs. However, the *fine-grained information* (e.g. textual explanations for the labels) is often ignored, and the systems do not explicitly generate the human-readable explanations. To solve this problem, we propose a novel generative explanation framework that learns to make classification decisions and generate fine-grained explanations at the same time. More specifically, we introduce the explainable factor and the minimum risk training approach that learn to generate more reasonable explanations. We construct two new datasets that contain summaries, rating scores, and fine-grained reasons. We conduct experiments on both datasets, comparing with several strong neural network baseline systems. Experimental results show that our method surpasses all baselines on both datasets, and is able to generate concise explanations at the same time.

1 Introduction

Deep learning methods have produced state-of-the-art results in many natural language processing (NLP) tasks (Vaswani et al., 2017; Yin et al., 2018; Peters et al., 2018; Wang et al., 2018; Hancock et al., 2018; Ma et al., 2018). Though these deep neural network models achieve impressive performance, it is relatively difficult to convince people to trust the predictions of such neural networks since they are actually black boxes for human beings (Samek et al., 2018). For instance, if an essay scoring system only tells the scores of a given essay without providing explicit reasons, the users can hardly be convinced of the judgment. Therefore, the ability to explain the rationale is essential for a NLP system, a need which requires traditional NLP models to provide human-readable explanations.

In recent years, lots of works have been done to solve text classification problems, but just a few of them have explored the explainability of their systems (Camburu et al., 2018; Ouyang et al., 2018). Ribeiro et al. (2016) try to identify an interpretable model over the interpretable representation that is locally faithful to the classifier. Samek et al. (2018) use heatmap to visualize how much each hidden element contributes to the predicted results. Although these systems are somewhat promising, they typically do not consider fine-grained information that may contain information for interpreting the behavior of models. However, if a human being wants to rate a product, s/he may first write down some reviews, and then score or summarize some attributes of the product, like price, packaging, and quality. Finally, the overall rating for the product will be given based on the fine-grained information. Therefore, it is crucial to build trustworthy explainable text classification models that are capable of explicitly generating fine-grained information for explaining their predictions.

To achieve these goals, in this paper, we propose a novel generative explanation framework for text classification, where our model is capable of not only providing the classification predictions but also generating fine-grained information as explanations for decisions. The novel idea behind our hybrid generative-discriminative method is to explicitly capture the fine-grained information inferred from raw texts, utilizing the information to help interpret the predicted classification results and improve the overall performance. Specifically, we introduce the notion of an explainable factor and a minimum risk training method that learn to generate reasonable explanations for the overall predict results. Meanwhile, such a strategy brings strong connections between the explanations and predictions, which in return leads to better performance. To the best of our knowledge, we are the first to explicitly explain the predicted results by utilizing the abstractive generative fine-grained information.

In this work, we regard the summaries (texts) and rating scores (numbers) as the fine-grained information. Two datasets that contain these kinds of fine-grained information are collected to evaluate our method. More specifically, we construct a dataset crawled from a website called PCMag[1]. Each item in this dataset consists of three parts: a long review text for one product, three short text comments (respectively explains the property of the product from positive, negative and neutral perspectives) and an overall rating score. We regard the three short comments as fine-grained information for the long review text. Besides, we also conduct experiments on the Skytrax User Reviews Dataset[2], where each case consists of three parts: a review text for a flight, five sub-field rating scores (seat comfortability, cabin stuff, food, in-flight environment, ticket value) and an overall rating score. As for this dataset, we regard the five sub-field rating scores as fine-grained information for the flight review text.

Empirically, we evaluate our model-agnostic method on several neural network baseline models (Kim, 2014; Liu et al., 2016; Zhou and Wang, 2018) for both datasets. Experimental results suggest that our approach substantially improves the performance over baseline systems, illustrating the advantage of utilizing fine-grained information. Meanwhile, by providing the fine-grained information as explanations for the classification results, our model is an understandable system that is worth trusting. Our major contributions are three-fold:

- We are the first to leverage the generated fine-grained information for building a generative explanation framework for text classification, propose an explanation factor, and introduce minimum risk training for this hybrid generative-discriminative framework;
- We evaluate our model-agnostic explanation framework with different neural network architectures, and show considerable improvements over baseline systems on two datasets;
- We provide two new publicly available explainable NLP datasets that contain fine-grained information as explanations for text classification.

2 Task Definition and Notations

The research problem investigated in this paper is defined as: How can we generate fine-grained explanations for the decisions our classification model makes? To answer this question, we may first investigate what are good fine-grained explanations. For example, in sentiment analysis, if a product A has three attributes: i.e., quality, practicality, and price. Each attribute can be described as “HIGH” or “LOW”. And we want to know whether A is a “GOOD” or “BAD” product. If our model categorizes A as “GOOD” and it tells that the quality of A is “HIGH”, the practicality is “HIGH” and the price is “LOW”, we can regard these values of attributes as good explanations that illustrate why the model judges A to be “GOOD”. On the contrary, if our model produces the same values for the attributes, but it tells that A is a “BAD” product, we then think the model gives bad explanations. Therefore, for a given classification prediction made by the model, we would like to explore more on the fine-grained information that can explain why it comes to such a decision for the current example. Meanwhile, we also want to figure out whether the fine-grained information inferred from the input texts can help improve the overall classification performance.

We denote the input sequence of texts to be $S\{s_1, s_2, \ldots, s_{|S|}\}$, and we want to predict which category $y_i (i \in [1, 2, \ldots, N])$ the sequence S belongs to. At the same time, the model can also produce generative fine-grained explanations e_c for y_i.

3 Generative Explanation Framework

In this part, we introduce our proposed Generative Explanation Framework (GEF). Figure 1 illustrates the architecture of our model.

3.1 Base Classifier and Generator

A common way to do text classification tasks is using an Encoder-Predictor architecture (Zhang

[1] https://www.pcmag.com/

[2] https://github.com/quankiquanki/skytrax-reviews-dataset

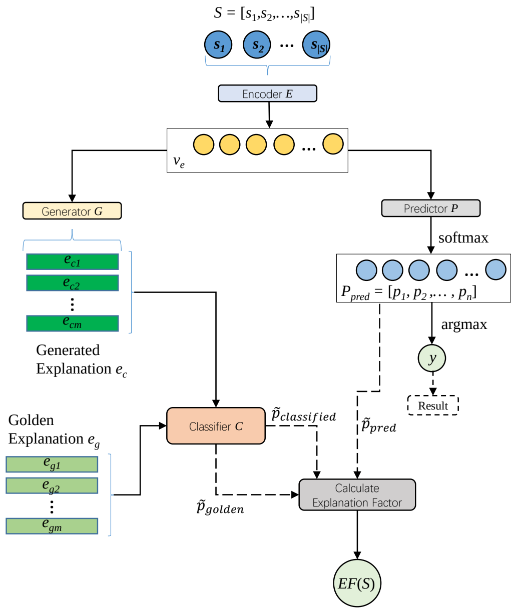

Figure 1: The architecture of the Generative Explanation Framework. E encodes S into a representation vector v_e. P gives the probability distribution P_{pred} for categories. We extract the ground-truth probability $\tilde{p}_{pred}$ from P_{pred}. Generator G takes v_e as input and generates explanations e_c. Classifier C and Predictor P both predict classes y. C will predict a probability distribution $P_{classified}$ when taking e_c as input, and predict P_{golden} when taking e_g as input, and then output the ground-truth probability $\tilde{p}_{classified}$ and $\tilde{p}_{golden}$. The explanation factor $EF(S)$ is calculated through $\tilde{p}_{pred}$, $\tilde{p}_{classified}$ and $\tilde{p}_{golden}$.

et al., 2015; Lai et al., 2015). As shown in Figure 1, a text encoder E takes the input text sequence S, and encodes S into a representation vector v_e. A category predictor P then gets v_e as input and outputs the category y_i and its corresponding probability distribution P_{pred}.

As mentioned above, a desirable model should not only predict the overall results y_i, but also provide generative explanations to illustrate why it makes such predictions. A simple way to generate explanations is to feed v_e to an explanation generator G to generate fine-grained explanations e_c. This procedure is formulated as:

$$v_e = Encoder([s_1, s_2, \cdots, s_{|S|}]) \tag{1}$$

$$P_{pred} = Predictor(v_e) \tag{2}$$

$$y = \arg\max_i (P_{pred,i}) \tag{3}$$

$$e_c = f_G(W_G \cdot v_e + b_G) \tag{4}$$

where $Encoder$ maps the input sequence $[s_1, s_2, \cdots, s_{|S|}]$ into the representation vector v_e; the $Predictor$ takes the v_e as input and outputs the probability distribution over classification categories by using the $softmax$.

During the training process, the overall loss $\mathcal{L}$ is composed of two parts, i.e., the classification loss $\mathcal{L}_p$ and explanation generation loss $\mathcal{L}_e$:

$$\mathcal{L}(e_g, S, \theta) = \mathcal{L}_p + \mathcal{L}_e \tag{5}$$

where θ represents all the parameters.

3.2 Explanation Factor

The simple supervised way to generate explanations, as demonstrated in the previous subsection, is quite straightforward. However, there is a significant shortcoming of this generating process: it fails to build strong connections between the generative explanations and the predicted overall results. In other words, the generative explanations seem to be independent of the predicted overall results. Therefore, in order to generate more reasonable explanations for the results, we propose to use an explanation factor to help build stronger connections between the explanations and predictions.

As we have demonstrated in the introduction section, fine-grained information will sometimes reflect the overall results more intuitively than the original input text sequence. For example, given a review sentence, “The product is good to use”, we may not be sure if the product should be rated as 5 stars or 4 stars. However, if we see that the attributes of the given product are all rated as 5 stars, we may be more convinced that the overall rating for the product should be 5 stars.

So in the first place, we pre-train a classifier C, which also learns to predict the category y by directly taking the explanations as input. More specifically, the goal of C is to imitate human beings’ behavior, which means that C should predict the overall results more accurately than the base model that takes the original text as the input. We prove this assumption in the experiments section.

We then use the pre-trained classifier C to help provide a strong guidance for the text encoder E, making it capable of generating a more informative representation vector v_e. During the training process, we first get the generative explanations e_c by utilizing the explanation generator G. We then feed this generative explanations e_c to the classifier C to get the probability distribution of the predicted results $P_{classified}$. Meanwhile, we can also get the golden probability distribution P_{gold} by feeding the golden explanations e_g to C. The process can be formulated as:

$$P_{classified} = softmax(f_C(W_C \cdot e_c + b_C)) \quad (6)$$

$$P_{gold} = softmax(f_C(W_C \cdot e_g + b_C)) \quad (7)$$

In order to measure the distance among predicted results, generated explanations and golden generations, we extract the ground-truth probability $\tilde{p}_{classified}$, $\tilde{p}_{pred}$, $\tilde{p}_{gold}$ from $P_{classified}$, P_{pred}, P_{gold} respectively. They will be used to measure the discrepancy between the predicted result and ground-truth result in minimum risk training.

We define our explanation factor $EF(S)$ as:

$$EF(S) = |\tilde{p}_{classified} - \tilde{p}_{gold}| + |\tilde{p}_{classified} - \tilde{p}_{pred}| \quad (8)$$

There are two components in this formula.

- The first part $|\tilde{p}_{classified} - \tilde{p}_{gold}|$ represents the distance between the generated explanations e_c and the golden explanations e_g. Since we pre-train C using golden explanations, we hold the view that if similar explanations are fed to C, similar predictions should be generated. For instance, if we feed a golden explanation “Great performance” to the classifier C and it tells that this explanation means “a good product”, then we feed another explanation “Excellent performance” to C, it should also tell that the explanation means “a good product”. For this task, we hope that e_c can express the same or similar meaning as e_g.

- The second part $|\tilde{p}_{classified} - \tilde{p}_{pred}|$ represents the relevance between the generated explanations e_c and the original texts S. The generated explanations should be able to interpret the overall result. For example, if the base model predicts S to be “a good product”, but the classifier tends to classify e_c to be the explanations for “a bad product”, then e_c cannot properly explain the reason why the base model gives such predictions.

3.3 Minimum Risk Training

In order to remove the disconnection between fine-grained information and input text, we use Minimum risk training (MRT) to optimize our models, which aims to minimize the expected loss, i.e., risk over the training data (Ayana et al., 2016). Given a sequence S and golden explanations e_g, we define $\mathcal{Y}(e_g, S, \theta)$ as the set of predicted overall results with parameter θ. We define $\Delta(y, \tilde{y})$ as the semantic distance between predicted overall results y and ground-truth $\tilde{y}$. Then, the objective function is defined as:

$$\mathcal{L}_{MRT}(e_g, S, \theta) = \sum_{(e_g,S)\in D} \mathbb{E}_{\mathcal{Y}(e_g,S,\theta)} \Delta(y, \tilde{y}) \quad (9)$$

where D presents the whole training dataset.

In our experiment, $\mathbb{E}_{\mathcal{Y}(e_g,S,\theta)}$ is the expectation over the set $\mathcal{Y}(e_g, S, \theta)$, which is the overall loss in Equation 5. And we define Explanation Factor $EF(S)$ as the semantic distance of input texts, generated explanations and golden explanations. Therefore, the objective function of MRT can be further formalized as:

$$\mathcal{L}_{MRT}(e_g, S, \theta) = \sum_{(e_g,S)\in D} \mathcal{L}(e_g, S, \theta) EF(S) \quad (10)$$

MRT exploits $EF(S)$ to measure the loss, which learns to optimize GEF with respect to the specific evaluation metrics of the task. Though $\mathcal{L}_{MRT}$ can be 0 or close to 0 when $\tilde{p}_{classified}$, $\tilde{p}_{pred}$ and $\tilde{p}_{gold}$ are close, this cannot guarantee that generated explanations are close to the golden explanations. In order to avoid the total degradation of loss, we define our final loss function as the sum of MRT loss and explanation generation loss:

$$\mathcal{L}_{final} = \sum_{(e_g,S)\in D} \mathcal{L} + \mathcal{L}_{MRT} \quad (11)$$

We try different weighting scheme for the overall loss, and get best performance with 1 :1.

3.4 Application Case

Generally, the fine-grained explanations are in different forms for a real-world dataset, which means that e_c can be in the form of texts or in the form of numerical scores. We apply GEF to both forms of explanations using different base models.

3.4.1 Case 1: Text Explanations

To test the performance of GEF on generating text explanations, we apply GEF to Conditional Variational Autoencoder (CVAE) (Sohn et al., 2015). We here utilize CVAE because we want to generate explanations conditioned on different emotions (positive, negative and neural) and CVAE

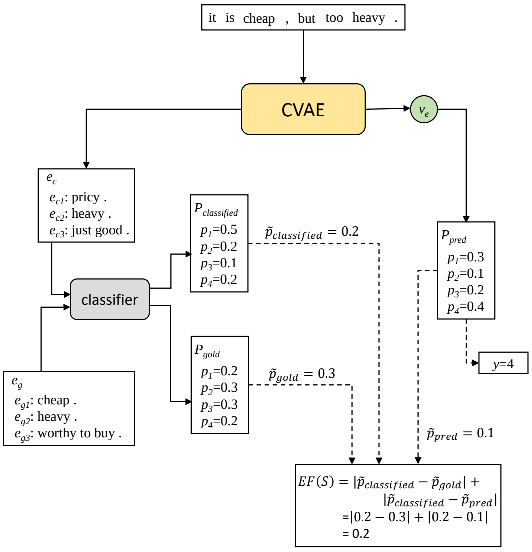

Figure 2: Structure of CVAE+GEF. There are totally 4 categories for the classification, and the ground-truth category is 2 in this example. We assume that the pre-trained classifier is a ”perfect” classifier that will correctly predict the final label to be 2 when taking e_g as input. So we wish the classifier can also predict the final result as label 2 when taking e_c as input. This is why we focus on $\tilde{p}_{classified}$ and $\tilde{p}_{gold}$.

is found to be capable of generating emotional texts and capturing greater diversity than traditional SEQ2SEQ models.

We give an example of the structure of CVAE+GEF in Figure 2. For space consideration, we leave out the detailed structure of CVAE, and will elaborate it in the supplementary materials. In this architecture, golden explanations e_g and generated explanations e_c are both composed of three text comments: positive comments, negative comments, and neutral comments, which are fine-grained explanations for the final overall rating. The classifier is a skip-connected model of bidirectional GRU-RNN layers (Felbo et al., 2017). It takes three kinds of comments as inputs, and outputs the probability distribution over the predicted classifications.

3.4.2 Case 2: Numerical Explanations

Another frequently employed form of the fine-grained explanations for the overall results is numerical scores. For example, when a user wants to rate a product, s/he may first rate some attributes of the product, like the packaging, price, etc. After rating all the attributes, s/he will give an overall rating for the product. So we can say that the rating for the attributes can somewhat explain why the user gives the overall rating. LSTM and CNN are shown to achieve great performance in text classification tasks (Tang et al., 2015), so we use LSTM and CNN models as the encoder E respectively. The numerical explanations are also regarded as a classification problem in this example.

4 Dataset

We conduct experiments on two datasets where we use texts and numerical ratings to represent fine-grained information respectively. The first one is crawled from a website called PCMag, and the other one is the Skytrax User Reviews Dataset. Note that all the texts in the two datasets are pre-processed by the Stanford Tokenizer[3] (Manning et al., 2014).

4.1 PCMag Review Dataset

This dataset is crawled from the website PCMag. It is a website providing reviews for electronic products, like laptops, smartphones, cameras and so on. Each item in the dataset consists of three parts: a long review text, three short comments, and an overall rating score for the product. Three short comments are summaries of the long review respectively from positive, negative, neutral perspectives. An overall rating score is a number ranging from 0 to 5, and the possible values that the score could be are $\{1.0, 1.5, 2.0, ..., 5.0\}$.

Since long text generation is not what we focus on, the items where review text contains more than 70 sentences or comments contain greater than 75 tokens are filtered. We randomly split the dataset into 10919/1373/1356 pairs for train/dev/test set. The distribution of the overall rating scores within this corpus is shown in Table 1.

4.2 Skytrax User Reviews Dataset

We incorporate an airline review dataset scraped from Skytraxs Web portal. Each item in this dataset consists of three parts: i.e., a review text, five sub-field scores and an overall rating score. The five sub-field scores respectively stand for the user’s ratings for seat comfortability, cabin stuff, food, in-flight environment, and ticket value, and each score is an integer between 0 and 5. The overall score is an integer between 1 and 10.

Similar to the PCMag Review Dataset, we filter out the items where the review contains more than

[3] https://nlp.stanford.edu/software/tokenizer.html

Overall Score	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
Number	21	60	283	809	2399	3981	4838	1179	78

Table 1: Distribution of examples by each overall rating score in PCMag Review Dataset.

Overall Score	1	2	3	4	5	6	7	8	9	10
Number	4073	2190	1724	1186	1821	1302	2387	3874	4008	4530

Table 2: Distribution of examples by each overall rating score in Skytrax User Reviews Dataset.

	Embedding	hidden	batch size
PCMag	GloVe, 100	128	32
Skytrax	random, 100	256	64

Table 3: Experimental settings for our experiments. Note that for CNN, we additionally set filter number to be 256 and filter sizes to be $[3, 4, 5, 6]$.

		BLEU-1	BLEU-2	BLEU-3	BLEU-4
Pos.	CVAE	36.1	13.5	3.7	2.2
	CVAE+GEF	**40.1**	**15.6**	**4.5**	**2.6**
Neg.	CVAE	33.3	14.1	3.1	2.2
	CVAE+GEF	**35.9**	**16.0**	**4.0**	**2.9**
Neu.	CVAE	30.0	8.8	2.0	1.2
	CVAE+GEF	**33.2**	**10.2**	**2.5**	**1.5**

Table 4: BLEU scores for generated explanations. Pos., Neg., Neu. respectively stand for positive, negative and neural explanations. The low BLEU-3 and BLEU-4 scores are because the target explanations contain many domain-specific words with low frequency, which makes it hard for the model to generate accurate explanations.

	Acc% (Dev)	Acc% (Test)
CVAE	42.07	42.58
CVAE+GEF	**44.04**	**43.67**
Oracle	46.43	46.73

Table 5: Classification accuracy on PCMag Review Dataset. *Oracle* means if we feed ground-truth text explanations to the Classifier C, the accuracy C can achieve to do classification. *Oracle* confirms our assumption that explanations can do better in classification than the original text.

300 tokens. Then we randomly split the dataset into 21676/2710/2709 pairs for train/dev/test set. The distribution of the overall rating scores within this corpus is shown in Table 2.

5 Experiments and Analysis

5.1 Experimental Settings

As the goal of this study is to propose an explanation framework, in order to test the effectiveness of proposed GEF, we use the same experimental settings on the base model and on the base model+GEF. We use GloVe (Pennington et al., 2014) word embedding for PCMag dataset and minimize the objective function using Adam (Kingma and Ba, 2014). The hyperparameter settings for both datasets are listed in Table 3. Meanwhile, since the generation loss is larger than classification loss for text explanations, we stop updating the predictor after classification loss reaches a certain threshold (adjusted based on dev set) to avoid overfitting.

5.2 Experimental Results

5.2.1 Results of Text Explanations

We use BLEU (Papineni et al., 2002) scores to evaluation the quality of generated text explanations. Table 4 shows the comparison results of explanations generated by CVAE and CVAE+GEF.

There are considerable improvements on the BLEU scores of explanations generated by CVAE+GEF over the explanations generated by CVAE, which demonstrates that the explanations generated by CVAE+GEF are of higher quality. CVAE+GEF can generate explanations that are closer to the overall results, thus can better illustrate why our model makes such a decision.

In our opinion, the generated fine-grained explanations should provide the extra guidance to the classification task, so we also compare the performance of classification on CVAE and CVAE+GEF. We use top-1 accuracy and top-3 accuracy as the evaluation metrics for the performance of classification. In Table 5, we compare the results of CVAE+GEF with CVAE in both test and dev set. As shown in the table, CVAE+GEF has better classification results than CVAE, which indicates that the fine-grained information can really help enhance the overall classification results.

As aforementioned, we have an assumption that if we use fine-grained explanations for classifica-

	s%	c%	f%	i%	t%
LSTM	46.59	52.27	43.74	41.82	45.04
LSTM+GEF	**49.13**	**53.16**	**46.29**	**42.34**	**48.25**
CNN	46.22	51.83	44.59	43.34	46.88
CNN+GEF	**49.80**	**52.49**	**48.03**	**44.67**	**48.76**

Table 6: Accuracy of sub-field numerical explanations on Skytrax User Reviews Dataset. s, c, f, t, v stand for seat comfortability, cabin stuff, food, in-flight environment and ticket value, respectively.

tion, we shall get better results than using the original input texts. Therefore, we list the performance of the classifier C in Table 5 to make the comparison. Experiments show that C has better performance than both CVAE and CVAE+GEF, which proves our assumption to be reasonable.

5.2.2 Results of Numerical Explanations

In the Skytrax User Reviews Dataset, the overall ratings are integers between 1 to 10, and the five sub-field ratings are integers between 0 and 5. All of them can be treated as classification problems, so we use accuracy to evaluate the performance.

The accuracy of predicting the sub-field ratings can indicate the quality of generated numerical explanations. In order to prove that GEF can help generate better explanations, we show the accuracy of the sub-field rating classification in Table 6. The 5 ratings evaluate the seat comfortability, cabin stuff, food, in-flight environment, and ticket value, respectively. As we can see from the results in Table 6, the accuracy for 5 sub-field ratings all get enhanced comparing with the baseline. Therefore, we can tell that GEF can improve the quality of generated numerical explanations.

Then we compare the result for classification in Table 7. As the table shows, the accuracy or top-3 accuracy both get improved when the models are combined with GEF.

Moreover, the performances of the classifier are better than LSTM (+GEF) and CNN (+GEF), which further confirms our assumption that the classifier C can imitate the conceptual habits of human beings. Leveraging the explanations can provide guidance for the model when doing final results prediction.

5.3 Human Evaluation

In order to prove our model-agnostic framework can make the basic model generate explanations more closely aligned with the classification results, we employ crowdsourced judges to evaluate a random sample of 100 items in the form of text, each being assigned to 5 judges on the Amazon Mechanical Turk. All the items are correctly classified both using the basic model and using GEF, so that we can clearly compare the explainability of these generated text explanations. We report the results in Table 8, and we can see that over half of the judges think that our GEF can generate explanations more related to the classification results. In particular, for 57.62% of the tested items, our GEF can generate better or equal explanations comparing with the basic model.

In addition, we show some the examples of text explanations generated by CVAE+GEF in Table 11. We can see that our model can accurately capture some key points in the golden explanations. And it can learn to generate grammatical comments that are logically reasonable. All these illustrate the efficient of our method. We will demonstrate more of our results in the supplementary materials.

	Acc%	Top-3 Acc%
LSTM	38.06	76.89
LSTM+GEF	**39.20**	**77.96**
CNN	37.06	76.85
CNN+GEF	**39.02**	**79.07**
Oracle	45.00	83.13

Table 7: Classification accuracy on Skytrax User Reviews Dataset. *Oracle* means if we feed ground-truth numerical explanation to the Classifier C, the accuracy C can achieve to do classification.

	Win%	Lose%	Tie%
CVAE+GEF	51.37	42.38	6.25

Table 8: Results of human evaluation. Tests are conducted between the text explanations generated by basic CVAE and CVAE+GEF.

5.4 Error and Analysis

We focus on the deficiency of generation for text explanation in this part.

First of all, as we can see from Table 11, the generated text explanation tend to be shorter than golden explanations. It is because longer explanations tend to bring more loss, so GEF tends to leave out the words that are of less informative, like function words, conjunctions, etc. In order to solve this problem, we may consider adding length reward/penalty by reinforcement learning to control the length of generated texts.

Product and Overall Rating	Explanations
Monitor, 3.0	**Positive Generated:** very affordable. unique and ergonomic design. good port selection.
	Positive Golden: unique design. dual hdmi ports. good color quality. energy efficient.
	Negative Generated: relatively faint on some features. relatively high contrast ratio. no auto port.
	Negative Golden: expensive. weak light grayscale performance. features are scarce.
	Neutral Generated: the samsung series is a unique touch-screen monitor featuring a unique design and a nice capacitive picture, but its color and grayscale performance could be better.
	Neutral Golden: the samsung series is a stylish 27-inch monitor offering good color reproduction and sharp image quality. however, it 's more expensive than most tn monitors and has a limited feature set.

Table 9: Examples of our generated explanations. Some key points are underlined.

Second, there are ⟨UNK⟩s in the generated explanations. Since we are generating abstractive comments for product reviews, there may exist some domain-specific words. The frequency of these special words is low, so it is relatively hard for GEF to learn to embed and generated these words. A substituted way is that we can use copy-mechanism (Gu et al., 2016) to generate these domain-specific words.

6 Related Work

Our work is closely aligned with Explainable Artificial Intelligence (Gunning, 2017), which is claimed to be essential if users are to understand, and effectively manage this incoming generation of artificially intelligent partners. In artificial intelligence, providing an explanation of individual decisions has attracted attention in recent years. The traditional way of explaining the results is to build connections between the input and output, and figure out how much each dimension or element contributes to the final output. Some previous works explain the result in two ways: evaluating the sensitivity of output if input changes and analyzing the result from a mathematical perspective by redistributing the prediction function backward (Samek et al., 2018). There are some works connecting the result with the classification model. Ribeiro et al. (2016) selects a set of representative instances with explanations via submodular optimization. Although the method is promising and mathematically reasonable, they cannot generate explanations in natural forms. They focus on how to interpret the result.

Some of the previous works have similar motivations as our work. Lei et al. (2016) rationalize neural prediction by extracting the phrases from the input texts as explanations. They conduct their work in an extractive way, and focus on rationalizing the predictions. However, our work aims not only to predict the results but also to generate abstractive explanations, and our framework can generate explanations both in the forms of texts and numerical scores. Hancock et al. (2018) proposes to use a classifier with natural language explanations that are annotated by human beings to do the classification. Our work is different from theirs in that we use the natural attributes as the explanations which are more frequent in reality. Camburu et al. (2018) proposes e-SNLI[4] by extending SNLI dataset with text explanations. And their simple but effective model proves the feasibility of generating text explanations for neural classification models.

7 Conclusion

In this paper, we investigate the possibility of using fine-grained information to help explain the decision made by our classification model. More specifically, we design a Generative Explanation Framework (GEF) that can be adapted to different models. Minimum risk training method is applied to our proposed framework. Experiments demonstrate that after combining with GEF, the performance of the base model can be enhanced. Meanwhile, the quality of explanations generated by our model is also improved, which demonstrates that GEF is capable of generating more reasonable explanations for the decision.

Since our proposed framework is model-agnostic, we can combine it with other natural processing tasks, e.g. summarization, extraction, which we leave to our future work.

[4] The dataset is not publicly available now. We would like to conduct further experiments on this dataset when it is released.

References

Shiqi Shen Ayana, Zhiyuan Liu, and Maosong Sun. 2016. Neural headline generation with minimum risk training. *arXiv preprint arXiv:1604.01904*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9539–9549. Curran Associates, Inc.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625.

J Gu, Z Lu, H Li, and VOK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Annual Meeting of the Association for Computational Linguistics (ACL), 2016*. Association for Computational Linguistics.

David Gunning. 2017. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web*.

Braden Hancock, Paroma Varma, Stephanie Wang, Martin Bringmann, Percy Liang, and Christopher Ré. 2018. Training classifiers with natural language explanations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1884–1895. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *EMNLP*.

Diederik P Kingma and Jimmy Lei Ba. 2014. Adam: A method for stochastic optimization. In *Proc. 3rd Int. Conf. Learn. Representations*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pages 2267–2273.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. *EMNLP*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of International Joint Conference on Artificial Intelligence*.

Xuezhe Ma, Zecong Hu, Jingzhou Liu, Nanyun Peng, Graham Neubig, and Eduard Hovy. 2018. Stack-pointer networks for dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1403–1414. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Sixun Ouyang, Aonghus Lawlor, Felipe Costa, and Peter Dolog. 2018. Improving explainable recommendations with synthetic reviews. *arXiv preprint arXiv:1807.06978*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM.

Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. 2018. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *ITU Journal: ICT Discoveries - Special Issue 1 - The Impact of Artificial Intelligence (AI) on Communication Networks and Services*, 1(1):39–48.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, Lisbon, Portugal. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Wei Wang, Ming Yan, and Chen Wu. 2018. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1705–1714.

Qingyu Yin, Yu Zhang, Weinan Zhang, Ting Liu, and William Yang Wang. 2018. Deep reinforcement learning for chinese zero pronoun resolution. *ACL*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Xianda Zhou and William Yang Wang. 2018. Mojitalk: Generating emotional responses at scale. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Victoria, Australia. ACL.

Supplemental Material

Structure of CVAE

By extending the SEQ2SEQ structure, we can easily get a Conditional Variational Antoencoder (CVAE) (Sohn et al., 2015; Zhou and Wang, 2018). Figure 3 shows the structure of the model.

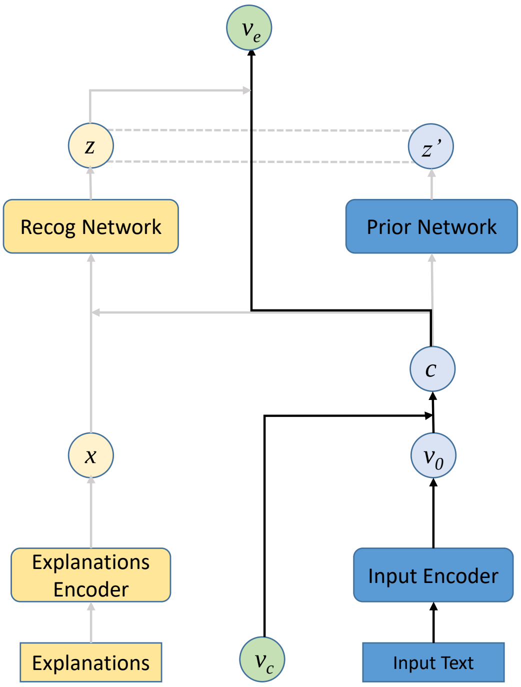

Figure 3: The structure of CVAE. The Input Encoder encodes the input text in v_0, and v_c is the control signal that determines the kind of fine-grained information (positive, negative and neutral). v_e is the initial input for the decoder. The Explanations Encoder encodes the short comment in x. Recognition Network takes x as input and produces the latent variable z. In our experiment, the Recognition Network and the Prior Network are both MLPs, and we use bidirectional GRU as the Explanations Encoder and Input Encoder.

To train CVAE, we need to maximize a variational lower bound on the conditional likelihood of x given c, where x and c are both random variables. In our experiment,$c = [v_c; v_0]$, and x is the text explanations we want to generate. This can be rewritten as:

$$p(x|c) = \int p(x|z,c)p(z|c)dz \qquad (12)$$

z is the latent variable. The decoder is used to approximate $p(x|z,c)$, denoted as $p_D(x|z,c)$, and Prior Network is used to approximate $p(z|c)$, denoted as $p_P(z|c)$. In order to approximate the true posterior $p(z|x,c)$, we introduce Recognition Network $q_R(z|x,c)$. According to Sohn et al. (2015), we can have the lower bound of $\log p(x|c)$ as:

Overall		s	c	f	i	t
9.0	**pred**	4.0	5.0	5.0	4.0	5.0
	gold	4.0	5.0	5.0	4.0	4.0
6.0	**pred**	3.0	5.0	3.0	3.0	4.0
	gold	4.0	5.0	3.0	3.0	4.0
2.0	**pred**	2.0	1.0	2.0	2.0	2.0
	gold	2.0	2.0	1.0	2.0	2.0

Table 10: Examples from the results on Skytrax User Reviews Dataset. s, c, f, i, t stand for seat comfortability, cabin stuff, food, in-flight environment and ticket value, respectively.

$$\begin{aligned} -\mathcal{L}(x,c;\theta) &= KL(q_R(z|x,c)||p_P(z|c)) \\ &\quad - \mathbb{E}_{q_R(z|x,c)}(\log p_D(x|z,c)) \qquad (13) \end{aligned}$$

θ is the parameters in the network. Notice that during training, z is used to train z' and passed to the decoder, but during testing, the ground truth explanations are absent and z' is passed to the decoder.

Output Sample

In this part, we provide some samples from our experiment.

Numerical Explanation Cases

We provide some numerical explanation cases in Table 10.

Text Explanation Cases

We provide some text explanation cases in Table 11.

Product and Overall Rating	Explanations
Television, 4.0	**Positive Generated:** Good contrast. Good black levels. Affordable.
	Positive Golden: Gorgeous 4k picture. Good color accuracy. Solid value for a large uhd screen.
	Negative Generated: Mediocre black levels. Poor shadow detail. Poor off-angle viewing.
	Negative Golden: Mediocre black levels. Poor input lag. Colors run slightly cool. Disappointing online features. Poor off-angle viewing.
	Neutral Generated: A solid, gorgeous 4k screen that offers a sharp 4k picture, but it's missing some features for the competition.
	Neutral Golden: A solid 4k television line, but you can get an excellent 1080p screen with more features and better performance for much less.
Flash Drive, 3.0	**Positive Generated:** Simple, functional design. Handy features.
	Positive Golden: Charming design. Reasonably priced. Capless design.
	Negative Generated: All-plastic construction. No usb or color protection.
	Negative Golden: All-plastic construction. On the slow side. Crowds neighboring ports. flash drives geared toward younger children don't have some sort of password protection.
	Neutral Generated: The tween-friendly ⟨UNK⟩ colorbytes are clearly designed and offers a comprehensive usb 3.0, but it's not as good as the competition.
	Neutral Golden: The kid-friendly dane-elec sharebytes value pack drives aren't the quickest or most rugged flash drives out there, but they manage to strike the balance between toy and technology. Careful parents would be better off giving their children flash drives with some sort of password protection.
TV, 4.0	**Positive Generated:** excellent picture. attractive glass-backed screen. hdr10 and dolby vision.
	Positive Golden: excellent picture with wide color gamut. stylish glass-backed screen. hdr10 and dolby vision. two remotes.
	Negative Generated: very expensive.
	Negative Golden: very expensive.
	Neutral Generated: lg's new oledg7p series is a stylish, attractive, and attractive hdtv line that's a bit more but not much more attractive.
	Neutral Golden: lg's signature oledg7p series is every bit as attractive and capable as last year's excellent oledg6p series, but the company has a new flagship oled that's only slightly more expensive but a lot more impressive.
Gaming, 4.0	**Positive Generated:** best-looking mainline pokemon game for the nintendo 3ds and feel. date, breathing, and dlc.
	Positive Golden: best-looking mainline pokemon game to date. alola trials mix up and vary progression over the gym-and-badge system, breathing new life into the game for longtime fans. ride pagers improve overworld navigation.
	Negative Generated: starts out very slow.
	Negative Golden: starts out very slow.
	Neutral Generated: the newest pokemon generation of sun/moon for the nintendo 3ds, making the feeling of the nintendo 3ds and remixes enough ideas to new life over making any wild, polarizing changes to the formula.
	Neutral Golden: the newest pokemon generation, sun/moon for the nintendo 3ds, tweaks and polishes the series' core concepts and remixes enough ideas to feel fresh without making any wild , polarizing changes to the formula.
Desktop, 3.5	**Positive Generated:** adjustable bulb. attractive design. energy efficient.
	Positive Golden: compact all in one. $500 price point. lenovo utilities. dynamic brightness system and eye distance system. no bloatware.
	Negative Generated: limited stand. no keyboard or micro between mac.
	Negative Golden: low power on benchmark tests. no usb 3.0. no hdmi. no video in or out. only 60-day mcafee anti-virus. camera is " always on. ".
	Neutral Generated: the lenovo thinkcentre edge is a good choice in the attractive design, and a few attractive colors in the price. it has a little bit of the best.
	Neutral Golden: the lenovo c325 is a good choice for those looking to spend only about $500 for a fully featured desktop pc. it's bigger than a laptop, and has the power to serve your web surfing and basic pc needs.

Table 11: Text examples from our generated explanations. ⟨UNK⟩ stands for "unknown word".